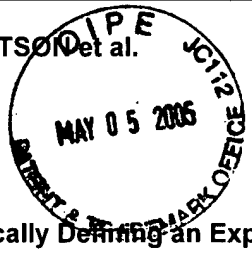
TRANSMITTAL FORM	Attorney Docket No. GB919990081US1/1751P



In re the application: **John B. IBBOTSON et al.** Confirmation No.: **8913**

Serial No: **09/675,468** Group Art Unit: **2173**

Filed: **September 28, 2000** Examiner: **Pillai, Namitha**

For: **Method and Tool for Graphically Defining an Expression**

ENCLOSURES *(check all that apply)*					
☐	Amendment/Reply	☐	Assignment and Recordation Cover Sheet	☐	After Allowance Communication to Group
	☐ After Final	☐	Part B-Issue Fee Transmittal	☐	Notice of Appeal
☐	Information disclosure statement	☐	Letter to Draftsman	■	(Revised) Appeal Brief
	☐ Form 1449	☐	Drawings	☐	Status Letter
	☐ (X) Copies of References	☐	Petition	■	Postcard
☐	Extension of Time Request *	☐	Fee Address Indication Form	■	Other Enclosure(s) *(please identify below):*
☐	Express Abandonment	☐	Terminal Disclaimer	- Response to Notice of Non-Compliant Appeal Brief	
☐	Certified Copy of Priority Doc	☐	Power of Attorney and Revocation of Prior Powers		
☐	Response to Incomplete Appln	☐	Change of Correspondence Address		
☐	Response to Missing Parts	*Extension of Term: Pursuant to 37 CFR 1.136, Applicant petitions the Commissioner to extend the time for response for xxxxxx month(s), from to .			
	☐ Executed Declaration by Inventor(s)				

CLAIMS					
FOR	Claims Remaining After Amendment	Highest # of Claims Previously Paid For	Extra Claims	RATE	FEE
Total Claims	19	20	0	$ 50.00	$ 0.00
Independent Claims	4	4	0	$200.00	$ 0.00
				Total Fees	$ 0.00

METHOD OF PAYMENT	
☐	Check no. _____ in the amount of $ _____ is enclosed for payment of fees.
☐	Charge $ _____ to Deposit Account No. _____ (Account Holder Name) for payment of fees.
■	Charge any additional fees or credit any overpayment to Deposit Account No. 09-0460 (IBM Corporation).

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT	
Attorney Name	Joseph A. Sawyer, Jr., Reg. No. 30,801
Signature	
Date	May 3, 2005

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on **May 3, 2005**

Type or printed name	Jimmy Nguyen
Signature	

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

APPEAL NO:

In Re Application of: Ibbotson, et al.

Serial No: 09/675,468

Filed: September 28, 2000

For: METHOD AND TOOL FOR GRAPHICALLY DEFINING AN EXPRESSION

APPELLANT'S REVISED APPEAL BRIEF

Joseph A. Sawyer, Jr.
Attorney for Appellants
International Business Machines Corporation
Sawyer Law Group, LLP
2465 E. Bayshore Road, Suite 406
Palo Alto, CA 94303

CERTIFICATE OF MAIL

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on **May 3, 2005**.

Jinny Nguyen

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In Re Application of:	Date: May 3, 2005
John B. IBBOTSON et al.	Confirmation No.: 8913
Serial No.: 09/675,468	Group Art Unit: 2173
Filed: September 28, 2000	Examiner: Pillai, Namitha

For: METHOD AND TOOL FOR GRAPHICALLY DEFINING AN EXPRESSION

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S REVISED BRIEF ON APPEAL

Sir:

Appellant herein files a Revised Appeal Brief in response to the Notice of Non-Compliant Appeal Brief dated April 4, 2005. Because the original Appeal Brief was filed June 28, 2004, prior to the effective date of 37 CFR 41.37, this Revised Appeal Brief is submitted in accordance with the provisions of former 37 C.F.R. § 1.192(c) as follows:

I. REAL PARTY IN INTEREST

Appellant respectfully submits that the above-captioned application is assigned, in its entirety to International Business Machines, of Armonk, New York.

II. RELATED APPEALS AND INTERFERENCES

Appellant states that, upon information and belief, he is not aware of any co-pending appeal or interference which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-9 and 11-20 are pending in the present application and stand rejected. Claims 1, 12, 13, 14, and 18-20 were amended, and claim 10 canceled in an Amendment dated June 2, 2003. Claims 1, 7-9, and 18-20 were amended again in an amendment dated November 14, 2003. Accordingly, Claims 1-9, and 11-20 are on appeal and all applied rejections concerning those claims are herein being appealed.

IV. STATUS OF AMENDMENT

All amendments have been entered.

V. SUMMARY OF THE INVENTION

The present invention is directed to a method and tool for graphically defining an expression, in particular, a query containing complex or non-native data types. (Specification, page 7, lines 2-18). According to the preferred embodiment of the present invention, the tool includes a graphic user interface (GUI) that allows a user to define a plurality of tree structures comprising a hierarchical series of nodes. (Specification, page 7, lines 7-8) At least one of the tree structures represents an input data structure and at least one other tree structure represents an output data structure. (Specification, page 17, lines 14-15). The user also is allowed to define

one or more lists that include a number of items, whereby each item on a list is associated with a node on a tree. (Specification, page 7, lines 8-10). Any item associated with the tree structure representing the input data structure is a filtering constraint and any item associated with the tree structure representing the output data structure defines a formatting definition. (Specification, page 17, lines 14-15).

Once the user has defined the plurality of tree structures, an expression generator analyzes the tree/list structure and generates an expression. (Specification, page 7, lines 12-13). The expression is used to configure modules in a relational message broker or to configure a database query. (Specification, page 17, lines 4-7).

The present invention, as recited in claims 1 and 18, provide:

> 1. A tool for graphically defining an expression, said tool comprising:
> a graphic user interface (GUI) component comprising:
> means, responsive to user input, for generating a graphic definition of the expression by defining a plurality of tree structures comprising a hierarchical series of nodes, and one or more lists comprising a plurality of items, each list item being associated with a respective node of an associated tree structure, wherein at least one of the tree structures represents an input data structure and at least one other tree structure represents an output data structure wherein any associated list item defines a formatting definition;
> an expression generator component adapted to read the graphic definition of the expression provided by a user through said GUI component, expression generator component comprising:
> means for analyzing said graphic definition and generating an expression based on the structure of each tree and any list items associated with respective nodes of a tree.
>
> 18. A method for graphically defining an expression in accordance with a graphic definition comprising the steps of:
> (a) defining a plurality of tree structures comprising a hierarchical series of nodes, and one or more lists comprising a plurality of items responsive to user input, each list item being associated with a respective node of an associated tree structure, wherein at least one of the tree structures represents an input data structure and at least one other tree structure represents an output data structure wherein any associated list item defines a formatting definition;
> (b) analyzing said definition; and

(c) generating an expression based on the structure of each tree and any list items associated with respective nodes of a tree.

Claims 19 and 20 are computer product and system claims, respectively, having similar scopes to that of claim 18.

VI. ISSUES

The issue presented is:

1. Whether claims 1, 4-6, 9, 11-15, and 18-20 are unpatentable under 35 U.S.C. §102(e) as being anticipated by Mizoguchi et al. (U.S. Patent No. 6,243,858).

2. Whether claims 2, 3 and 16 were properly rejected under 35 U.S.C. §103(a) as being unpatentable over Mizoguchi in view of MacLeod et al (U.S. Patent No. 6,434,545).

3. Whether claim 7 was properly rejected under 35 U.S.C. §103(a) as being unpatentable over Mizoguchi and Premerlani et al. (U.S. Patent No. 5,555,367).

4. Whether claim 8 was properly rejected as being unpatentable over Mizoguchi and Lee et al. (U.S. Patent No. 6,535,883).

5. Whether claim 17 was properly rejected as being unpatentable over Mizoguchi in view of MacLeod and further in view of Moshfeghi (U.S. Patent No. 6,476,833).

VII. GROUPING OF CLAIMS

Appellants hereby state that claims 1-9 and 11-20 do not stand or fall together. Appellant hereby states that claims 1-9, and 11-17 form one group and claims 18-20 form a second group.

VIII. ARGUMENTS

A. Summary of the Applied Rejections

In the Final Office Action, the Examiner rejected claims 1, 4-6, 9, 11-15, and 18-20 under 35 U.S.C. §102(e) as being anticipated by Mizoguchi et al. (U.S. Patent No. 6,243,858). The Examiner rejected claims 2, 3 and 16 under 35 U.S.C. §103(a) as being unpatentable over Mizoguchi in view of MacLeod et al (U.S. Patent No. 6,434,545). Claim 7 was rejected under 35 U.S.C. §103(a) as being unpatentable over Mizoguchi and Premerlani et al. (U.S. Patent No. 5,555,367). Claim 8 was rejected as being unpatentable over Mizoguchi and Lee et al. (U.S. Patent No. 6,535,883). Claim 17 was rejected as being unpatentable over Mizoguchi in view of MacLeod and further in view of Moshfeghi (U.S. Patent No. 6,476,833).

In rejecting the independent claims, the Examiner stated:

> Referring to claims 1 and 18-20, Mizoguchi discloses a tool for graphically defining an expression with a graphic user interface (GUI) component with means for responding to user input for generating a graphic definition of the expression by defining plurality of tree structures (Figures 6 and 11), wherein as seen by Figure 11, the tree structures represent a distinct nodes under which are represented items which are defined by the user (reference numbers S52 and S53, Figure 10). As also seen by the structures of Figure 11, wherein "input data 1" comprise a hierarchical series of nodes the nodes being represented under "input data 1" and "output data 1", thereby showing a hierarchical relationship. These tree structures also contain a plurality of times listed with the list item being associated with the respective nodes of the associated tree, wherein the nodes and the items both represented as the same data, thereby showing an association are disclosed as being associated with the respective input or output trees. Figure 11 of Mizoguchi also clearly discloses an input data structure and at least one other tree structure representing an output data structure wherein any associated list item defines a formatting definition, used for the graphic expression. Mizoguchi discloses an expression generator component adapted to read the graphic definition of the expression provided by a user through the GUI component, with the expression generator analyzing the graphic definition and generating an expression based on the structure of each tree and any list items associated with respective nodes of a tree (Figure 23), wherein the grid representation which is the graphic definition is executed and the result of the execution is outputted, this involving analyzing and generating of an expression from the graphic representation.

Final Office Action, pp. 2-3.

B. The Cited Prior Art

Mizoguchi is directed to a tool that assists a software developer in creating a business application program. In Mizoguchi, various basic program modules constituting a business application program are prepared beforehand and made available when the tool is utilized. When an application program is actually constructed, module tiles representing the individual program modules are displayed in the form of a module palette. At the same time, a grid panel on which the module tiles are arranged is displayed. The user arranges the module tiles necessary for the business application program on the grid panel and the tool builds the application program from the user's arrangement. (Abstract).

MacLeod is related to a graphical query analyzer. A query is displayed and represented as a tree with each operation in the execution plan represented by a tree node. The analyzer computes the cost of each operation and when a user selects an operation (node) displays costs statistics related to the selected operation. (Abstract). Premerlani is directed to a system for generating computer programs for queries by manipulating object-oriented diagrams. (Abstract). Lee is directed to a graphical user interface for creating a set of validation rules. (Abstract). Moshfeghi is directed to a system for browsing markup language documents from within the context of a client-server application running on an end-user device. (Abstract).

C. Independent Claims 1 and 18-20 Are Allowable.

Appellants respectfully submit that Mizoguchi fails to teach or suggest generating a graphic definition of an expression by "defining a plurality of tree structures comprising a hierarchical series of nodes, and one or more lists comprising a plurality of items, each list item being associated with a respective node of an associated tree structure, wherein at least one of the tree structures represents an input data structure and at least one other tree structure represents an output data structure wherein any associated list item defines a formatting definition," as recited

in claims 1, 18-20. Thus, Appellants respectfully submit that claims 1 and 18-20 are allowable over Mizoguchi.

1. Mizoguchi fails to teach or suggest defining "a plurality of tree structures," as recited in claims 1, and 18-20.

Appellants respectfully submit that Mizoguchi does not teach or suggest defining "a plurality of tree structures," as recited in claims 1, and 18-20. In Mizoguchi, the tool allows the user to connect a chain of program modules to create a graphic representation of an application program. Each program module performs various types of data processing, for example update processing, retrieval processing, data manipulation processing, input processing and print processing. (Col. 7, lines 39-50).

Only one chain of modules is created. (See for example, Figures 6A-6C, 8, 13 and 25). Accordingly, Mizoguchi teaches defining one chain or tree structure representing an application program. Contrary to the present invention, Mizoguchi does not teach or suggest defining "a plurality of tree structures," as recited in claims 1, and 18-20, and certainly does not teach or suggest defining tree structures having the properties recited in claims 1 and 18-20.

In the Final Office Action, the Examiner indicates that the plurality of tree structures is taught in Figures 6 and 11 of Mizoguchi, and that "wherein as seen by Figure 11, the tree structures represent a distinct nodes under which are represented items which are defined by the user." Figures 6A-6C illustrate display screens during the application program creation process where module tiles are selected and laid out. Figure 11 illustrates a screen for giving a data record processing definition of a module tile. In both sets of Figures, each program module (tile) is a distinct node in a tree structure that represents the application program. Thus, in Figures 6A-6C and 11, Mizoguchi teaches defining only one tree structure and not *a plurality* of tree structures, as recited in independent claims 1, and 18-20.

2. Mizoguchi fails to teach or suggest "a plurality of tree structures comprising a hierarchical series of nodes," as recited in claims 1, and 18-20.

Even if Mizoguchi were to be interpreted as teaching a plurality of tree structures, where each program module might be construed to a "tree structure," Appellants respectfully submit that the program modules do not comprise "a hierarchical series of nodes," as recited in claims 1, and 18-20. As stated above, the program modules perform various types of data processing (col. 2, lines 1-2; col. 7, lines 39-50). Intuitively, a program module comprises lines of programming code that instruct the module to process input data according to the module's purpose. Nothing teaches or suggests that any of the program modules is a tree structure "comprising a hierarchical series of nodes," as recited in claims 1, and 18-20.

If, on the other hand, each program module is construed to be a plurality of tree structures represented by the input data record list, the processing list, and the output data record item list respectively (as shown in item 60 of Figure 11), then Appellants again respectfully submit that such lists fail to teach or suggest "a plurality of tree structures comprising a hierarchical series of nodes," as recited in claims 1, and 18-20.

Rather, Figure 11 depicts the "data record processing definition screen 60" associated with the "retrieval processing" program module. This screen 60 is displayed to the user after he or she has completed defining input and output *data records* for the module (steps S4 and S5 of Figure 5). In the definition screen 60, "the input data record items are listed in a processing definition frame of input data records and the output data record items are listed in a processing definition frame of output data records. The description of the processing of output data record items by reference to the input data record items can be written in the PROCESSING area." (Col. 10, lines 1-7).

Appellants respectfully submit that Figure 11 does not teach or suggest one tree structure

or a plurality of tree structures "comprising a *hierarchical* series of nodes" but rather, it displays lists of input and output data record items and what processing, if any, was performed on each input record item to produce the corresponding output data item. While each item on the input list *is transformed* to a corresponding item on the output list (via the processing), nothing teaches or suggests that the relationship between the items on the input list and the items on the output list is *hierarchical*, or that the relationship *between* the items on any particular list is hierarchical. For example, referring to Figure 11, *output* data item 2 is related mathematically to *input* data item 2 by the function, "input data item 2+100." This is *not* a hierarchical relationship. Also, nothing in Mizoguchi teaches or suggests any relationship, let alone a hierarchical relationship, between record items on the input or output lists.

3. Mizoguchi fails to teach or suggest defining "one or more lists comprising a plurality of items, each list item being associated with a respective node of an associated tree," as recited in claims 1, and 18-20.

Appellants respectfully submit that Mizoguchi fails to teach or suggest "one or more lists comprising a plurality of items, *each list item being associated with a respective node of an associated tree*," as recited in claims 1, 18-20. If the program module is represented by a tree structure and each list in the definition screen 60 is considered a node (which is disputed above), then, contrary to the present invention, each node is associated with *several* items on a list.

4. Mizoguchi fails to teach or suggest "an output data structure wherein any associated list item defines a formatting definition," as recited in claims 1 and 18-20.

Finally, Appellants respectfully submit that nothing in Mizoguchi in general and in particular Figure 11 teaches or suggests "an output data structure wherein any associated list item defines *a formatting definition*," as recited in claims 1 and 18-20. In the Final Office Action, the Examiner relies on Figure 11 to teach this feature where the output data record item list is

construed to be the present invention's "output data structure." Nevertheless, as stated above, the output data record item list fails to teach or suggest a tree structure "comprising a hierarchical series of nodes." Even if it did, nothing teaches or suggests that the output data record item on the list "defines a formatting definition." Indeed, according to Mizoguchi, the definition of the data record means "the *name* of the data record itself . . . and the *names* of the items included in the data record." (Col. 8, lines 53-57). Nothing teaches or suggest that the *name* of the output data record item "defines a formatting definition," as recited in claims 1 and 18-20.

Accordingly, based on the arguments above, Applicants respectfully submit that Mizoguchi fails to teach or suggest the cooperation of elements recited in claims 1, 18, 19 and 20. Thus, claims 1, 18, 19 and 20 are allowable over the cited reference.

Claims 4-6, 9, and 11-15 depend on claim 1, and the above arguments apply with full force. Therefore, Appellants respectfully submit that claims 4-6, 9, and 11-15 are also allowable over the cited reference.

D. Dependent Claims 2, 3, 7, 8, 16 and 17 Are Allowable

Claims 2, 3, 7, 8, 16 and 17 depend on claim 1 and therefore, the arguments with regard to claim 1 apply with full force to claims 2, 3, 7, 8, 16 and 17. Thus, even if the secondary references disclose the features described by the Examiner, claims 2, 3, 7, 8, 16 and 17 are still allowable because Mizoguchi fails to teach or suggest the present invention as recited in claim 1. Accordingly, Appellants respectfully submit that claims 2, 3, 7, 8, 16 and 17 are allowable over the cited references.

E. Summary of Arguments

For the reasons set forth above, Appellants respectfully submit that claims 1-9 and 11-20

are allowable over the cited references. Appellants respectfully request that the final rejection of claims 1-9 and 11-20 be reversed.

Note: For convenience of detachment without disturbing the integrity of the remainder of pages of this Appeal Brief, Appellant's APPENDIX A is attached on separate sheets following the signatory portion of this Appeal Brief.

This Appeal Brief is being submitted in triplicate, and authorization for payment of the required Appeal Brief fee is contained in the cover letter for this brief. Please charge any fee that may be necessary for the continued pendency of this application to Deposit Account No. 50-0563 (IBM Corporation).

Respectfully submitted,

SAWYER LAW GROUP LLP

May 3, 2005
Date

Joseph A. Sawyer, Jr.
Attorneys for Appellants
Reg. No. 30,801
(650) 493-4540

IX. APPENDIX A

1. (amended) A tool for graphically defining an expression, said tool comprising:

a graphic user interface (GUI) component comprising:

means, responsive to user input, for generating a graphic definition of the expression by defining a plurality of tree structures comprising a hierarchical series of nodes, and one or more lists comprising a plurality of items, each list item being associated with a respective node of an associated tree structure, wherein at least one of the tree structures represents an input data structure and at least one other tree structure represents an output data structure wherein any associated list item defines a formatting definition;

an expression generator component adapted to read the graphic definition of the expression provided by a user through said GUI component, expression generator component comprising:

means for analyzing said graphic definition and generating an expression based on the structure of each tree and any list items associated with respective nodes of a tree.

2. (original): The tool according to claim 1 wherein said expression is adapted to configure one of a plurality of nodes of a relational message broker, a message broker or database query.

3. (original): The tool according to claim 2 wherein said expression is an SQL3 expression.

4. (original): The tool according to claim 1, where said nodes comprise leaf and branch nodes, said branch nodes representing complex structured fields and said leaf nodes representing simple

fields comprising one of a string, integer, real or a date.

5. (original): The tool according to claim 4 wherein each list item comprises an expression.

6. (original) The tool according to claim 5 wherein said GUI component is adapted to allow a user to define a tree structure representing an input data structure wherein any associated list item defines a filtering constraint.

7. (currently amended): The tool according to claim 1 wherein said GUI component is adapted to allow a user to define two or more input tree structures, each having an associated list, at least one list item associated with a first node of a first input tree structure identifying a second node of second input tree structure from which said expression generator generates an expression joining said two input tree structures on said nodes.

8. (amended): The tool according to claim 1 wherein said GUI component is adapted to allow a user to define an input tree structure having two or more associated lists, at least one list item from each list comprising an expression from which said expression generator generates a logical OR expression.

9. (amended): The tool according to claim 1 wherein said GUI component is adapted to allow a user to graphically link two or more nodes within one or more input tree structures from which said expression generator generates a logical expression limiting said nodes to equality.

10. (canceled)

11. (amended): The tool according to claim 1 wherein said GUI component is adapted to allow a user to define an input tree structure and an output tree structure, each having an associated list, at least one of said list items for said output tree structure identifying a node of said input tree structure.

12. (amended): The tool according to claim 1 wherein said GUI component is adapted to display a list for an output tree to the left of the tree.

13. (amended): The tool according to claim 1 wherein said GUI component is adapted to allow a user to define a list item comprising a free variable, said free variable representing the associated tree structure node within said graphical definition.

14. (amended): The tool according to claim 1 wherein said GUI component is adapted to allow a user to define a tree structure comprising a node represented by a wildcard symbol, said wildcard symbol representing said node and all otherwise undefined sub-structures of said node.

15. (original): The tool according to claim 14 wherein said GUI component is adapted to allow a user to define a structure comprising a branch node having a sub-structure comprising one or more defined nodes and a node represented by a wildcard symbol.

16. (original): The tool according to claim 1 wherein said analyzing means is cooperable with a grammatical definition of said graphic definition to generate said expression.

17. (original): The tool according to claim 2 wherein one of said nodes comprises a filter for filtering XML messages.

18. (amended): A method for graphically defining an expression in accordance with a graphic definition comprising the steps of:

(a) defining a plurality of tree structures comprising a hierarchical series of nodes, and one or more lists comprising a plurality of items, each list item being associated with a respective node of an associated tree structure, wherein at least one of the tree structures represents an input data structure and at least one other tree structure represents an output data structure wherein any associated list item defines a formatting definition;

(b) analyzing said graphic definition; and

(c) generating an expression based on the structure of each tree and any list items associated with respective nodes of a tree.

19. (amended): A computer readable medium containing program instructions for graphically defining an expression in accordance with a graphic definition, the program instructions for:

defining a plurality of tree structures comprising a hierarchical series of nodes, and one or more lists comprising a plurality of items, each list item being associated with a respective node of an associated tree structure, wherein at least one of the tree structures represents an input data structure and at least one other tree structure represents an output data structure wherein any associated list item defines a formatting definition;

analyzing said graphic definition; and

generating an expression based on the structure of each tree and any list items associated with respective nodes of a tree.

20. (amended): A system for graphically defining an expression in accordance with a graphic definition comprising:

means for defining a plurality of tree structures comprising a hierarchical series of nodes, and one or more lists comprising a plurality of items responsive to user input, each list item being associated with a respective node of an associated tree structure, wherein at least one of the tree structures represents an input data structure and at least one other tree structure represents an output data structure wherein any associated list item defines a formatting definition;

means for analyzing said graphic definition; and

means for generating an expression based on the structure of each tree and any list items associated with respective nodes of a tree.